Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

LISTING OF CLAIMS:

1. (Original) A method comprising:

receiving a user selection of a first instruction from a list of instructions that executed during a processor simulation; and

tracing an operand in the first instruction directly to a use of the operand in a second instruction in the list of instructions by following operand dependencies between such first and second instructions.

2. (Original) The method of claim 1 wherein tracing determines that the second instruction set the value of the operand as used in the first instruction as a source operand.

3. (Original) The method of claim 1 wherein tracing determines that a next use of the operand, after that of the first instruction as a destination operand, occurs in the second instruction.

4. (Original) The method of claim 1 wherein tracing comprises:

determining attributes of the first instruction; and

using the attributes of the first instruction to find the second instruction.

5. (Original) The method of claim 4 wherein receiving comprises:

receiving a selected cycle corresponding to the first instruction.

6. (Original) The method of claim 5 further comprising:

determining a program counter value associated with the selected cycle.

7. (Original) The method of claim 6 wherein determining attributes comprises:

using the program counter value to look up the attributes in an instruction operand map that provides attributes of each instruction, including instruction type and type of registers used by such instruction type for operands.

8. (Original) The method of claim 7 wherein the instructions are instructions of a microcode and the instruction operand map is generated at microcode build time

9. (Currently Amended) The method of claim 7 wherein using the program counter value to look up attributes comprises:

determining for each register type a physical address.

10. (Original) The method of claim 9 wherein determining the physical address comprises determining whether each register type is a non-I/O register or an I/O register.

11. (Original) The method of claim 10 wherein determining the physical address comprises determining whether each non-I/O register is accessed using an index register.

12. (Original) The method of claim 11 wherein the instruction operand map is used to provide the physical address for each non-I/O register that is not accessed using an index register.

13. (Original) The method of claim 11 wherein the physical address for each non-I/O register that is determined to be accessed using an index register is determined by obtaining a historical value of the index register at the selected cycle from a register history that records historical values of registers for each register type as such values change during simulation.

14 (Currently Amended) The method of claim 12 wherein the physical address ~~register~~ for any register determined to be an I/O register is obtained for the selected cycle from a memory reference history that records physical addresses and reference counts for each of the I/O registers that is used in a memory reference during simulation.

15. (Original) The method of claim 9 wherein determining the program counter value comprises looking up the program counter value in a program counter history that records state change events, which are detected during simulation, with associated program counter values for each cycle in which such state change events occurred.

16. (Original) The method of claim 15 wherein tracing further comprises: using the physical address for each register used in the first instruction to traverse the program counter history, instruction by instruction, to find a matching physical address in the second instruction.

17. (Original) The method of claim 16 wherein the microcode is intended for execution on one or more microengines in a processor simulated by the processor simulation and wherein the program counter history of more than one of the microengines is traversed.

18. (Original) The method of claim 1 wherein the instructions are intended for execution on at least one microengine of the processor simulated by the processor simulation.

19. (Currently Amended) The method of claim 18 wherein the microengine is configured to support multiple threads of execution and the instructions are ~~microcode is~~ intended for execution by at least one of the multiple threads of execution ~~threads~~.

20. (Currently Amended) An article comprising:

a storage medium having stored thereon instructions that when executed by a machine result in the following:

receiving a user selection of a first instruction from a list of instructions that executed during a processor simulation; and

tracing an operand used in the first instruction directly to a use of the operand in a second instruction in the list of instructions by following operand dependencies between such first and second instructions,

wherein tracing comprises:

determining attributes of the first instruction selected by the user; and

using the attributes of the first instruction selected by the user to find the second instruction,

wherein determining attributes comprises:

using a program counter value to look up the attributes in an instruction operand map that provides attributes of each instruction, including instruction type and type of registers used by such instruction type for operands and to determine for each type of register a physical address.

21. (Original) The article of claim 20 wherein tracing determines that the second instruction set the value of the operand as used in the first instruction.

22. (Original) The article of claim 20 wherein tracing determines that a next use of the operand after that of the first instruction occurs in the second instruction.

23. (Currently Amended) A storage medium having ~~graphical user interface~~ executable instructions stored and configured to be executed by a processor, the executable instructions comprising:

executable instructions to render a window having ~~showing~~ a view of microcode instructions that executed on a processor simulator during a simulation and for which a simulation history has been collected by the processor simulator;

wherein executable instructions to render a window having the view ~~being usable~~ comprises executable instructions to provide a tracing option in a menu presented to a user for one of the microcode instructions as an instruction of interest,~~; and~~ the tracing option being usable to trace any variable used by the instruction of interest in the simulation history directly to a second instruction in which a most recent change to or next use of such variable occurred,

the executable instructions further comprising executable instructions to:

receive a user selection of the instruction of interest; and

trace an operand in the instruction of interest directly to a use of the operand in the second instruction of the microcode instructions by following operand dependencies between the instruction of interest and the second instruction.

24. (Currently Amended) The storage medium ~~graphical user interface~~ of claim 23 wherein selection of the tracing option by the user causes a submenu of options available for the instruction of interest to be provided to the user, each of the options of the submenu corresponding to one of the variables used by the instruction of interest.

25. (Currently Amended) The storage medium ~~graphical user interface~~ of claim 23 ~~22~~ further comprising executable instructions to render:

a second window in which a cycle of interest corresponding to the instruction of interest is indicated;

wherein the indication of the cycle of interest is modified to indicate a new cycle of interest corresponding to the second instruction; and

wherein the ~~first~~ window is modified to reflect the new cycle of interest.

26. (Original) A device comprising:

at least one line card for forwarding networking data to ports of a switching fabric;

the at least one line card comprising a network processor comprising multi-threaded microengines each configured for execution with a microcode; and

wherein the microcode comprises a microcode developed using a debugger tool that allowed tracing of operands in code lines of the microcode once executed by a simulator simulating operation of the network processor.

27. (Currently Amended) The device ~~system~~ of claim 26 wherein the operands are associated with registers in the microengines, and the registers include general purpose registers and I/O transfer registers.